Dynamic vision : does 3D scene perception necessarily need two cameras or just one ?

Ming Xie

INRIA

UNITE DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 974

Programme 6

DYNAMIC VISION : DOES 3D SCENE PERCEPTION NECESSARILY NEED TWO CAMERAS OR JUST ONE ?

Ming XIE

Février 1989

RR_0974

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone : 99 36 20 00
Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

Publication Interne n° 448 - Janvier 1989 - 38 Pages

Dynamic Vision : Does 3D scene perception necessarily need two cameras or just one ?

===

Vision Dynamique : Perception 3D nécessite-t-elle d'avoir deux caméra ou seulement une ?

Ming XIE

INRIA/IRISA de Rennes
Campus de Beaulieu
F-35042 Rennes Cedex FRANCE
Tel (33)-99-36-20-00 poste 412

Abstract

Most of beings have two eyes to recognize their environment. But the basic principle of a vision system is "one eye + motion". In this paper, we first demonstrate a new proposition, that is :

> In machine vision, 3D scene perception needs necessarily two images, but not two cameras. The basic structure of a vision system is one camera with its motion (called "dynamic vision").

We will then give one application of this proposition to the reconstruction of a 3D polyhedral scene by a moving camera. This problem is decomposed into the following subproblems : (a) how to approximate a edge image by a set of 2D line segments; (b) how to match the set of 2D line segments within a sequence of images in order to obtain the dynamic parameters of 2D line segments; (c) how to recover the 3D line segments from the knowledge about the camera motion and the dynamic parameters of 2D line segments and (d) how to interpret the set of 3D line segments by a set of 3D polygons.

Key Words : Dynamic vision, Motion, Image corresponding velocities, Reconstruction of 3D primitives, Edge linking, Matching of 2D primitives, Interpretation of 3D primitives.

Résumé

En vue de percevoir leur environnement, beaucoup d'animaux ont deux yeux. Mais la vision de base est "un œil avec son mouvement". Dans ce rapport, nous allons d'abord démontrer une nouvelle proposition, à savoir :

> En vision par ordinateur, la perception d'une scène 3D a besoin nécessairement d'avoir deux images mais non pas deux caméra. La structure de base pour un système de vision est une caméra avec son mouvement (appellé "Vision Dynamique").

Ensuite nous appliquons cette proposition pour résoudre le problème de la reconstruction d'une scène 3D polyédrique par une caméra mobile. Nous décomposons ce problème en quatre sous-problèmes : (a) comment approximer polygonalement une image contour; (b) comment mettre en correspondance les segments 2D dans une séquence d'images pour estimer les paramètres dynamiques des segments 2D; (c) comment reconstriure les segments 3D à partir de la connaissance sur le mouvement de la caméra et sur les paramètres dynamiques des segments 2D et (d) comment interpréter un ensemble des segments 3D par des polygones 3D.

Key Words : Vision dynamique, Mouvement, Vitesse relative aux images, Reconstruction des primitives 3D, Approximation polygonale, Mise en correspondance des primitives 2D, Interprétation des primitives 3D.

Table des matières

1 Introduction

Computer vision is a science studing how to perceive and how to understand the visual information about an environment. Many techniques of perception have been already developped : monocular vision, stereovision, laser range finder and structured lighting detection, etc. Concerning the second subject of research, many of studies have been contributed to : shape from shading, shape from texture, shape from contour and shape from motion, etc. (see [HOR 77], [ALO et al. 85],[IKE et al. 81], [IKE 84], [WAN et al. 87], [BOL et al. 86], [LOW 87], etc)

Recently a new study has been investigated concerning the use of the camera motion to help the visual perception and how to merge the different perceptions corresponding to a sequence of images. This is known as the " Dynamic vision". If the motion is controllable, it is said to be the "Active vision". (see [ALO et al. 87], [RIV 87], [AYA 88]).

In the field of dynamic vision, a fundamental question is asked :"Does one really need to use two cameras or only one camera to realize the 3D scene perception?". The 3D scene means here a 3D static scene. We give our answer to this question by defining a new proposition, that is :

> In machine vision, 3D scene perception needs necessarily two images, but not two cameras. The basic structure of a vision system is one camera with its motion (called "dynamic vision").

In the world, almost all of the animals has two eyes to recognize their environment. But these two eyes do not work dependently. For exemple, the bird's two eyes cannot observe the same scene at the same time. Each one observes two different parts of its environment at a time. So each one perceives and understands independently the scene because each eye can give itself a motion necessary to the 3D scene perception! But one may ask why the animals do have two eyes? There are two reasons for that : (a) firstly two eyes can widen the fields of vision and (b) secondly two eyes allow to have two images for the 3D scene perception in the case of static observation. (see the figure 1).

3D perception by a moving eye is the basic principle for an animal's vision. This capability can also be realized in a machine like robot!.

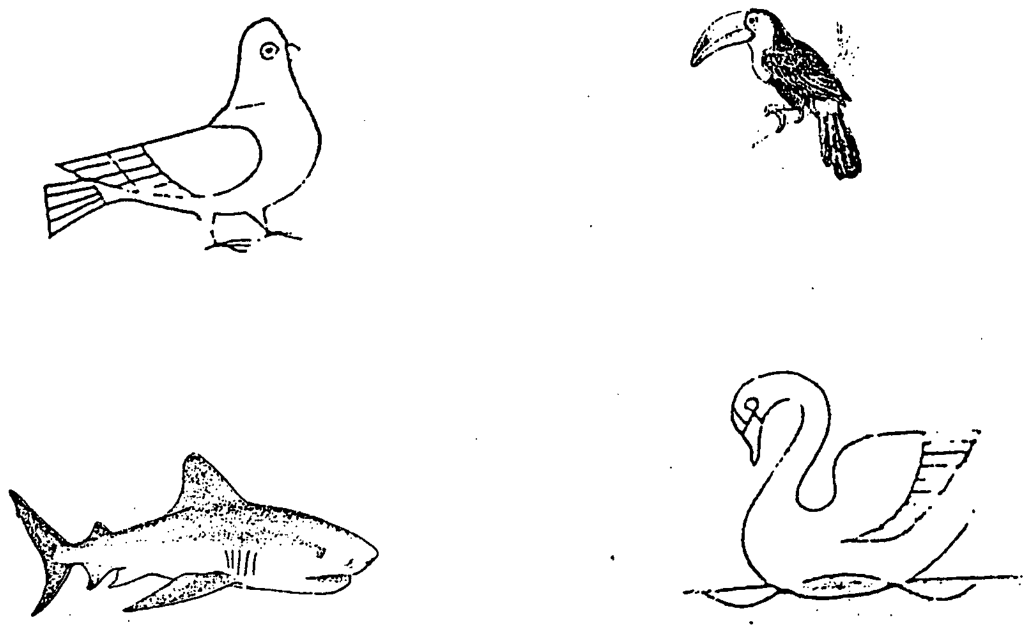


Figure 1 : Some animals whose two eyes do not observe the same scene at the same moment

In the next section, we will give a mathematical demonstration to reveal the principle of 3D primitives perception by a moving camera. In the third section, we will apply this principle to the problem of the reconstruction of 3D polyhedral scene by a moving camera; this section is divided into two parts : we first treat the problem of the perception of 3D line segments and then we deal with the problem of the understanding of 3D line segments. Our research works are realized in a dynamic vision system named VIDYR, that is presented in the fourth section. At the end of the paper, we give some results.

2 Principle of 3D primitives perception

Without a-priori knowledge about the scene, the perception of 3D static primitives necessarily needs two images which can be acquired by one camera moving between two different positions or by two cameras. In the later case, one can consider that the first camera moves to the position of second camera so that the second image is acquired. So two images can always be considered to be acquired by a moving camera at two different positions.

2.1 Motion vector: the representation of image corresponding velocities

The camera motion plays an important role in the 3D scene perception. Because the images are acquired at some discrete times, so we are just interested in the average motion velocities of camera between each two time samples. These motion velocities are called the *image corresponding velocities*. In the following presentation, the camera motion will mean the image corresponding velocities.

If we work in the camera's coordinate system, the camera motion is interpreted by the motion of the 3D static scene. In the following presentation, the motion will mean the motion of 3D static scene in the space of camera's coordinate system.

Considering a 3D point P moving during the time Δt from the position P_1 to P_2, its motion is composed of two parts : the translation component $t(t)$ and the rotation component $r(t)$. So the image corresponding velocities of the 3D point P is defined as (see [XIE et al. 88d]) :

$$\begin{cases} V_t &= T(t) \\ AS(V_r) &= \frac{1}{\sqrt{1+TR(R(t))}}[R(t)^T - R(t)] \end{cases} \tag{1}$$

with :

$$\begin{cases} T(t) &= \int_t^{t+\Delta t} t(u)\, du \\ \\ R(t) &= \prod_{u=0}^{\Delta t} r(t+\Delta t - u) \end{cases} \tag{2}$$

where (V_t, V_r) are respectively the translation velocity and the rotation velocity; $AS()$ means the anti-symmetric matrix of a vector; $TR()$ means the trace of a matrix and $R(t)^T$ is the transposed matrix of $R(t)$.

These equations tell us that the image corresponding velocities do not depend on the motion paths of camera but only on the discrete positions of camera.

If $X = \{X_1, X_2, X_3\}$ represents the coordinates of the point P in the camera's coordinate system, we have then :

$$\dot{X} = X(t+\Delta t) - X(t) = V_t + V_r \times X. \tag{3}$$

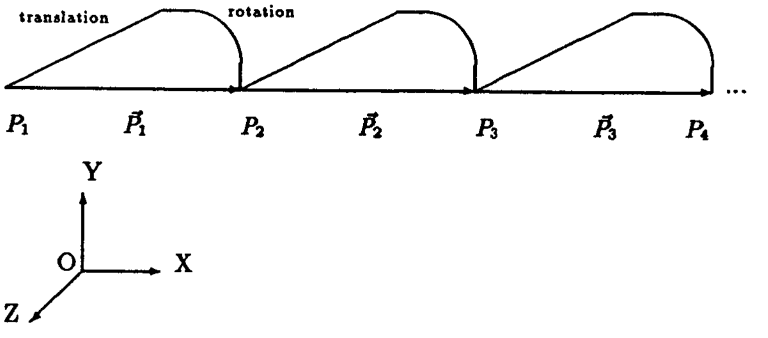


Figure 2 : Motion vector: $\vec{P_i} = P_i\vec{P}_{i+1}$ and the Speed Path: $P_1\vec{P}_2 + P_2\vec{P}_3 + P_3\vec{P}_4 + ...$

where $\dot{X}$ means the average derivative of X with respects to time.

In the figure 2, $\dot{X}$ is represented by the vector $\widehat{P_1P_2}$, we call it the *motion vector.* We can now define the concept about the *velocity path* which is the curve composed of the set of motion vectors. The velocity path is determined only by the discrete positions of camera at which the images are acquired. So different motion paths may have the same velocity path and the image corresponding velocities depend only on the velocity paths.

We will see later that it is the motion vector which gives the unique solution of the 3D scene perception.

2.2 Principle of triangulation

We use the perspective projection as the camera model (see the figure 3). A 3D point P will be projected to its image p according to the following relationship :

$$\begin{cases} x \times X = \vec{0} \\ x_3 = f \end{cases} \tag{4}$$

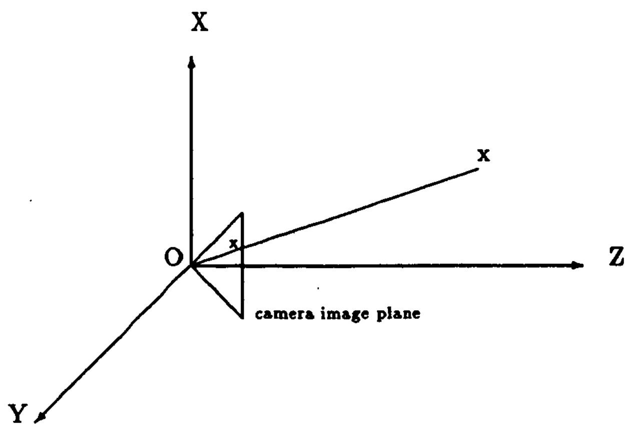


Figure 3 : Camera model : Perspective Projection

Where ($x = \{x_1, x_2, x_3\}, X = \{X_1, X_2, X_3\}$) are respectively the coordinates of an image point and of a 3D point and f is the camera's focal length.

We think of x as the 2D static primitive. From the above relationship, it is obvious that one cannot find P from the knowledge about p. This is why we have emphasized that the 3D perception needs necessarily two images, that allows to extract the 2D dynamic primitives : x with its derivative $\dot{x}$.

We give now a proposition about the principle of triangulation :

> The 3D geometrical primitive is only determined by its motion vector and its corresponding 2D dynamic images

Proof : We will demonstrate this proposition in two steps :

- Step I : Geometrical illustration for the triangulation of points :

  In the figure 4, first drawing the ray s from O to p and the ray t from O to $p + \dot{p}$; we know that the 3D points P and $P + \dot{P}$ can only be localized at the rays s and t because of the perpective projection of camera. Then we draw a subset of lines from the points in s to the

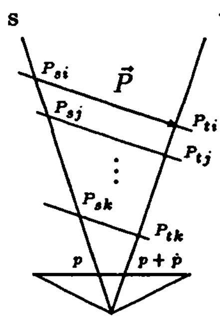


Figure 4 : Triangulation Principle

points in t. These lines must be parallel to the motion vector $\dot{P}$. We get in this way a subset of vector :

$$\{(\widehat{P_{si}P_{ti}}) \mid i = 0, 1, 2, ...\}$$

Among these vectors, it is obvious that there is only one vector which is equal to the motion vector : $\widehat{P_{si}P_{ti}} = \dot{P}$. So the position of the 3D point P is unique, that is : $P = P_{si}$ and $P + \dot{P} = P_{ti}$

- Step II : Theoretical Analysis :

We have given a geometrical illustration about the principle of triangulation in 2D space. Now we will make a theoretical analysis in the general case of triangulation. Before that, we first want to define the concept about a "geometrical primitive". All of the geometrical objects : line, plane, surfaces of higher order, etc; can be considered as a set of 3D points. So a geometrical primitive is defined as a subset of 3D points which can be characterized by a geometrical function like :

$$F(X, W) = 0.$$

where X is the coordinates of a 3D point and W the parameter vector. Then, an object is considered to be composed of some geometrical primitives.

After the motion with the image corresponding velocities (V_t, V_r), the geometrical function of a geometrical primitive becomes :

$$\begin{cases} dF(X, \dot{X}, W, \dot{W}) = 0 \\ \dot{X} = V_t + V_r \times X \end{cases} \tag{5}$$

That is :

$$dF(X, V_t, V_r, W, \dot{W}) = 0. \tag{6}$$

From the following equations :

$$\begin{cases} F(X, W) = 0 \\ dF(X, V_t, V_r, W, \dot{W}) = 0 \end{cases} \tag{7}$$

We get :

$$G(V_t, V_r, W, \dot{W}) = 0. \tag{8}$$

We call it as the **Characteristic Function of Geometry** about a geometrical primitive.

By the perspective projection of the camera, one can obtain its image of a geometrical primitive. This image can be characterized in the camera coordinate system by the following equations :

$$\begin{cases} f(x, w) = 0 \\ R(w, W) = 0 \end{cases} \tag{9}$$

And their derivatives :

$$\begin{cases} df(x, \dot{x}, w, \dot{w}) = 0 \\ dR(w, \dot{w}, W, \dot{W}) = 0 \end{cases} \tag{10}$$

We call the equation $dR() = 0$ as the **Intrinsical Relationship Function** about a geometrical primitive and its projection image.

According to the following equations :

$$\begin{cases} G(V_t, V_r, W, \dot{W}) = 0 \\ dR(w, \dot{w}, W, \dot{W}) = 0 \end{cases} \tag{11}$$

we see that the 3D parameters $(W, \dot{W})$ can only be determined by the motion parameters of the camera $(-V_t, -V_r)$ and the 2D dynamic parameters of the geometrical primitive projection $(w, \dot{w})$. It proves the proposition presented in the introduction section, that is to say that one must have just only a moving camera to realize the 3D scene perception.

3 Reconstruction of a 3D polyhedral scene

In this section, we will give one application of the proposition about the 3D scene perception. The goal is to recover a 3D polyhedral scene by a moving camera. We decompose the problem into two subproblems : (a) the perception of 3D line segments and (b) the interpretation of 3D line segments.

3.1 Modeling of a 3D polyhedral scene

In a polyhedral universe, we are interested in the 2D/3D edge informations, that is : the 3D polygons of a polyhedral object; the 3D line segments of a 3D polygon; the patterns in an edge image; the chains of a pattern and the 2D line segments approximating a chain. So a polyhedral scene can be effectively modeled by its 2D/3D edge informations. The figure 5 shows this modeling.

3.2 Perception of 3D line segments

The resolution of the problem is realized in three steps : (a) the polygonal approximation of an edge image; (b) the matching of 2D line segments and (c) the triangulation of 3D line segments.

3.2.1 Polygonal approximation of an edge map

The edge images are obtained by an algorithm developped according to the method presented in [DER 87]. We have then studied an efficient algorithm for the polygonal approximation of an edge image (see [XIE et al. 88b]).

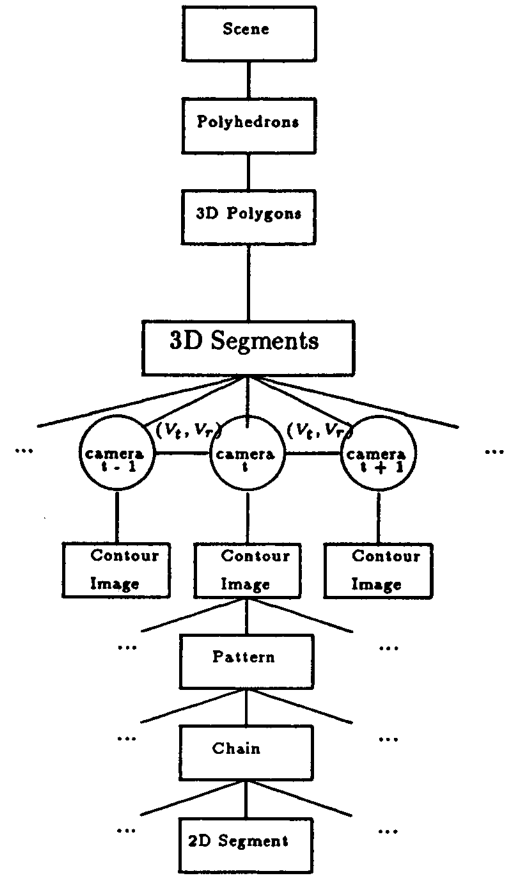


Figure 5 : Multi-level primitives in a polyhedral scene

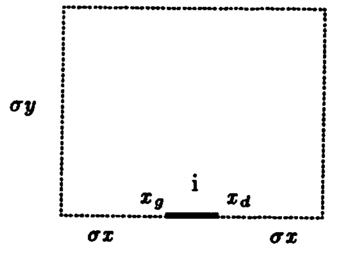


Figure 6 : Contour element i and its causal region of neighborhood

First, we resolve the problem about the edge linking. In our method, we have defined a contour element which is the subset of edge pixels localized consecutively in the same line of the edge image. In this way, the edge linking becomes the linking of the contour elements, that can be done more rapidly. With each contour element, we associate to it a causal region of neighborhood, that is :

$$\begin{cases} x \in [x_g - \sigma x, x_d + \sigma x] & if \quad y \in [y_c - \sigma y, y_c[\\ x \in [x_g - \sigma x, x_d] & if \quad y = y_c \end{cases} \tag{12}$$

where (x_g, x_d) are the X coordinates of the two end points of a contour element and y_c is its Y coordinate (see the figure 6).

By scanning from line to line the image of the contour elements, we take the following linking operations : the creature of the chains; the lengthening of the chains and the closure of the chains. The algorithm of contour element linking produces a set of simple chains that can be directly splitted into a subset of 2D line segments. That means that all chains do not have any branchs and the contour elements belonging to a chain are sequentially localized in the X and Y directions (see the figure 7). The parameters of the 2D line segments are estimated by the least square method.

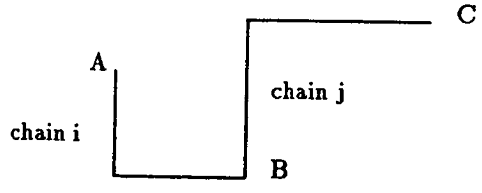


Figure 7 : According to the definition of the contour elements, the pattern $\widehat{ABC}$ is composed of two chains : $\widehat{AB}$ and $\widehat{CB}$

3.2.2 Matching of 2D line segments

The matching of 2D line segments within a sequence of images is a very difficult task. In [XIE et al. 88c], we have succesfully resolved this problem in two steps : (a) the temporal matching of 2D line segments and (b) the spatial matching of 2D line segments (see the figure 8).

In the temporal matching, we work at each instant with two images : the present image and the next image. The goal is to estimate the 2D line segments in the present image and to predict their corresponding 2D line segments in the next image.

First, we extract the contour in the present image and make the edge linking. With each edgel, we know its position and its orientation. Then we use a selected template presented in [BOU 85] to estimate its normal optical flow. Finally, from the informations about the edge linking and the edgels' normal optical flows, we can estimate directly the dynamic parameters of the 2D line segments by the least square method. We model a 2D line segment by its *plücke* coordinates (see [RIV 87]) : (m, l). So for each 2D line segment, we know the values about (m, l) and their derivatives $(\dot{m}, \dot{l})$. The 2D line

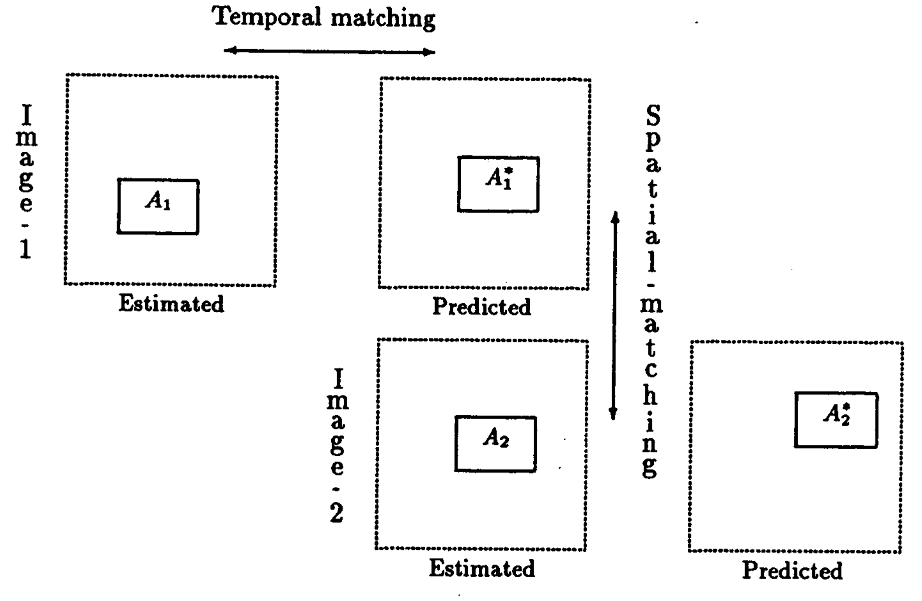


Figure 8 : Temporal and Spatial Matching of 2D line Segments

segments parameterized by $(m + \dot{m}, l + \dot{l})$ are called the *predicted* 2D line segments in the next images. The figure 9 illustrates the described operation.

Having resolved the temporal matching, the spatial matching becomes a trivial task. In each image, we know two sets of 2D line segments : (a) the set of 2D line segments estimated at the current instant and (b) the set of 2D line segments predicted at the preceding instant. The spatial matching consists of the matching of these two sets. This is done by finding the nearest neighbor among one set for a 2D line segment in the other set. In [XIE et al. 88c], we have defined a distance measure to determine the nearest neighbors. The figure 10 shows the combined matching of 2D line segments within a sequence of images.

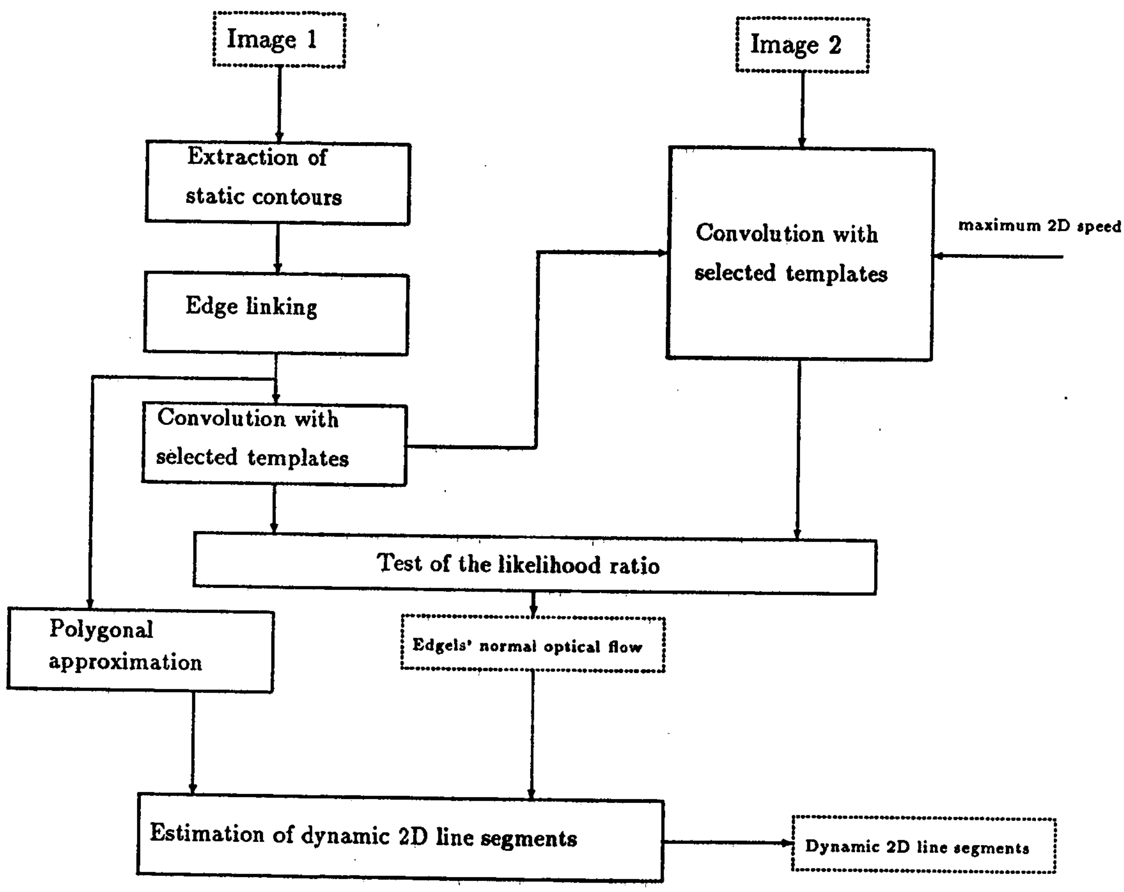


Figure 9 : Temporal matching : Estimation of dynamic 2D line segments

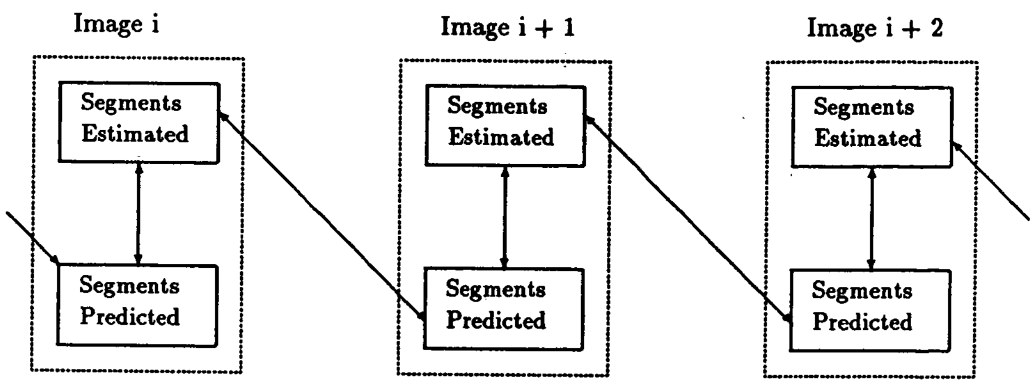


Figure 10 : Matching of 2D line segments in a sequence of images

3.2.3 Triangulation of 3D line segments

A 3D line segment is modeled by its *plücke* coordinates (see [RIV 87]) : (U, H), so its equation is written as :

$$X \times U = H. \tag{13}$$

And its Characteristic function of geometry is described by :

$$\begin{cases} \dot{U} &= V_r \times U. \\ \dot{H} &= V_r \times H + V_t \times U. \end{cases} \tag{14}$$

Where $(-V_t, -V_r)$ are the image corresponding velocities of the moving camera (see [RIV 87]).

A 3D line segment is projected generally to its corresponding 2D line segment that is also modeled by the *plücke* coordinates (m, l). So the equation of a 2D line segment is written as :

$$x \times m = l. \tag{15}$$

According to the perspective projection, it is easy to find the intrinsical relationship between a 3D line segment and its image (2D line segment) :

$$\begin{cases} \dot{l} \times H + l \times \dot{H} & = 0 \\ \dot{m}^t \bullet H + m^t \bullet \dot{H} & = 0 \\ \dot{l}^t \bullet U + l^t \bullet \dot{U} & = 0 \end{cases} \tag{16}$$

From the above equations, we can find the unique solutions about (U, H) :

1. Calculation of U :

$$\begin{cases} U = \frac{a \times l}{\|a \times l\|} \\ \\ a = \dot{l} + W^t \bullet l \end{cases} \tag{17}$$

2. Calculation of H :

$$\begin{cases} H = k \bullet (m \times l \times U) \\ \\ k = \frac{-L^t \bullet (S \bullet U)}{(L + W^t \bullet L)^t \bullet (m \times l \times U)} \\ \\ L = (l_3 - l_2, l_1 - l_3, l_2 - l_1)^t \end{cases} \tag{18}$$

Where (S, W) are the anti-symmetric matrix corresponding to the vectors V_t and V_r.

3.3 Interpretation of 3D line segments

It is assumed that the 3D line segments belong to the polyhedral scene. So it is possible to reconstruct the 3D polygons from a set of 3D line segments. A 3D polygon is a limited region in a 3D plane (called support plane). A 3D plane is generally parameterized by its normal vector N and its distance ρ to the origin of the coordinate system; its equation is written as :

$$N \bullet X = \rho. \tag{19}$$

So the identification of 3D polygons consists of :

1. how to estimate the normal vector N of the support planes.
2. how to estimate the parameters ρ of the support planes.
3. how to localize the polygons in their support planes.

The figure 11 illustrates the above three steps by an exemple of the identification of a block.

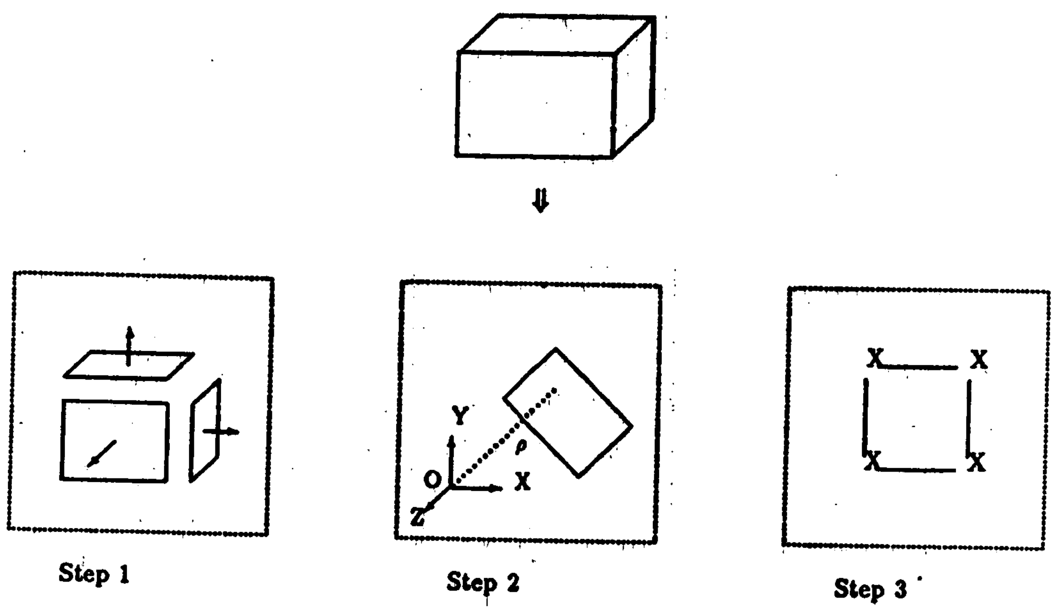


Figure 11 : The three steps for the identification of 3D polygons

3.3.1 Estimation of orientations of the support planes

Given a set of 3D line segments, we classify them into some subsets so that the 3D line segments in each subset form a group of parallel support planes. To do that, we have studied two efficient techniques (see [XIE et al. 88a]) : (a) the dynamic Prediction/Verification method and (b) the generalized HOUGH transform method.

Knowing that the subset $\{U_i \mid i = 1, 2, ..., k\}$ of 3D line segments belong to a group of parallel support planes, the normal vectors N of these planes can be estimated by the least square method, that means to find N minimizing the following criterion :

$$J(N) = \min \left[\sum_{i=1}^{k} (U_i \bullet N)\right]. \tag{20}$$

3.3.2 Estimation of the parameters ρ

For a given normal vector N, it may correspond to some parallel support planes having different parameters ρ. We first project the mid-points $\{M_i \mid i = 1, 2, ..., n\}$ of the 3D line segments (belonging to the corresponding subset) to the axis $\widehat{ON}$ passing through the vector N. In this way, we obtain some clouds of 1D points in the axis $\widehat{ON}$ (see the following figure).

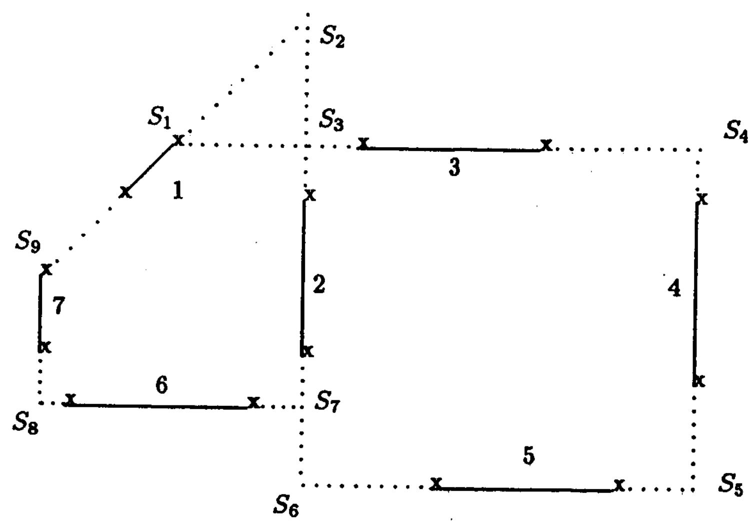
Axis $\widehat{ON}$

+++++ o oooooo xxxxxxxxx

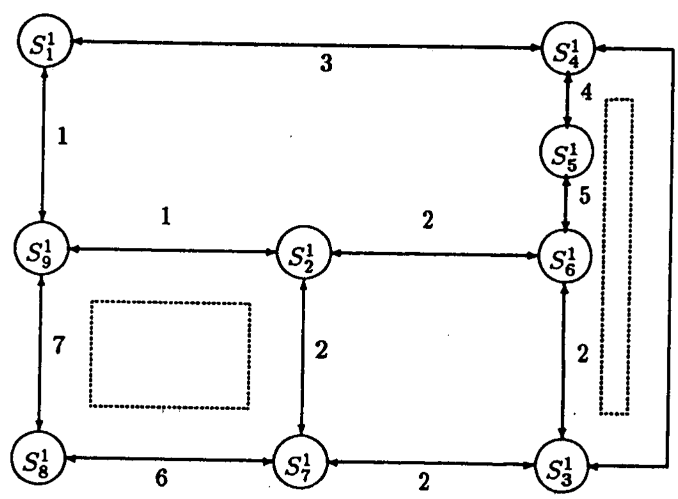
The centroids of these clouds correspond to the parameters ρ. We use a technique of the linking of 1D contour to localize these clouds in the axis $\widehat{ON}$ and to find the parameters ρ (see [XIE et al. 88a]).

3.3.3 Linking of 3D line segments

Given an estimated support plane, we need to know the existence of some polygons in it and their positions. For a subset of 3D line segments in a support plane, we may first build a neighborhood graph of these line segments and try to find some closing boundaries in this graph. If there are not any closing boundaries in a support plane, it means that no polygon exists in it. The figure 12 shows an exemple of the linking of 3D line segments by the neighborhood graph (see [XIE et al. 88a]).



(A) Estimated support plan



(B) Neighborhood graph

Figure 12 : An exemple of the linking of 3D line segments by the neighborhood graph. The dashed rectangles correspond to the real polygons. The numbers represent the line segments and the S_is represent the vertexs

4 Experimental results

We have developped a dynamic vision system VIDYR at IRISA. The architecture of this system is shown by the figure 13, which includes :

1. The control unit of VIDYR :
   It is the central unit of the system VIDYR, that manages the communications of the data : the parameters, the results and the motion paths; and also the addition of a new application into VIDYR.

2. The simulation subsystem VISYR :
   This simulation tool has been developped at IRISA (see [HEG 87]). The 3D scene is described by a CAO language LGRC. The synthesis of images is based on an algorithm of "ray casting". This simulation tool allows to have a sequence of synthetic images corresponding to the selected motion path.

3. The subsystem of a moving camera :
   A subsystem of the robot arm AID has been developped at IRISA (see [CHA et al. 88]). We have implanted a CCD camera at its wrist that is connected to the image processing system EDIXIA. The camera can move with the robot arm by following a selected motion path so that a sequence of images can be acquired.

4. The graphic window of dialog :
   The host of the system VIDYR is a Sun workstation under UNIX. We have used the functions of the *SunView** and the *SunCore** to create graphic window that allows : (a) the graphic interface between the system VIDYR and the users, and (b) the graphic visualization of the results.

We are developping the algorithms presented in this paper in the system VIDYR. We give some results below (the resolution of image is 256×256) :

1. The figure 15 shows an exemple about the polygonal approximation of the edge image. There are 3331 edgels in the contour image and 2256 corresponding contour elements. 47 chains are created, that are approximated by 74 2D line segments. The execution time is below 6 second of *cpu* and the size of the region of neighborhood are ($\sigma x = 1, \sigma y = 2$).

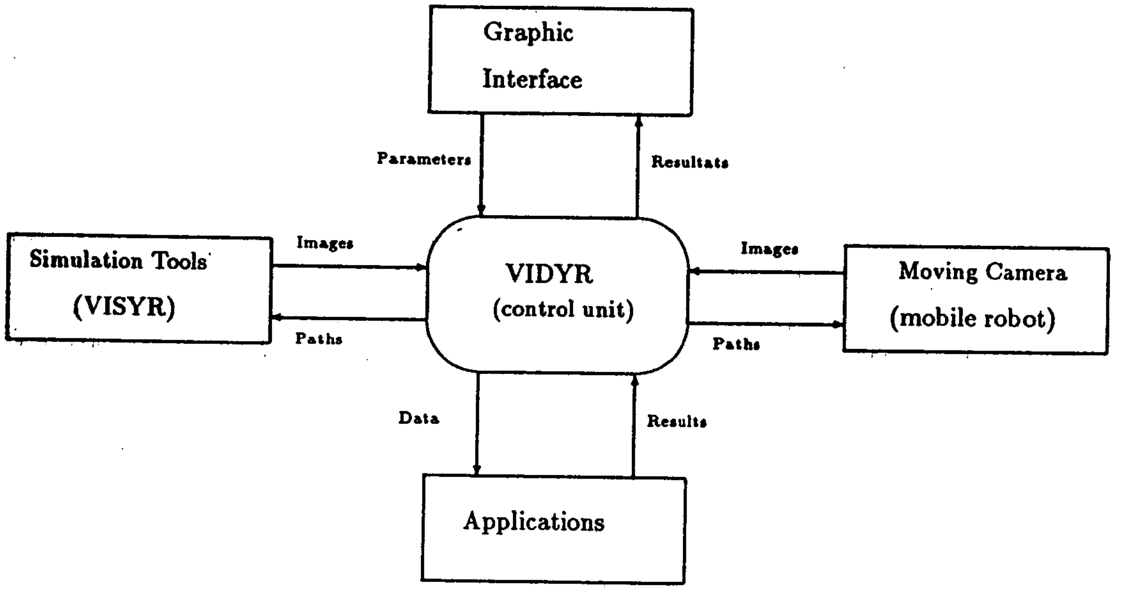


Figure 13 : Dynamic Vision System: VIDYR

2. The figure 16 shows an exemple about the temporal matching of 2D line segments. The maximum 2D velocity is set to be ± 11 pixels. The total execution time is below 68 seconds of *cpu*, including about 40 seconds of *cpu* used by the contour extraction.

3. The figure 17 shows an exemple about the temporal/spatial matching of 2D line segments. We can find that the short 2D line segments and the 2D line segments approximating a curve miss to be matched. The execution time is below 8 seconds of *cpu*.

4. The figure 19 shows an exemple about the triangulation of 3D line segments. This algorithm is currently tested with real images.

5. The figure 20 shows an exemple about the identification of 3D polygons from a set of 3D line segments. The variances of orientations of the support planes are below 8 degree and the variances of the parameters ρ are below $10cm$. The exeucution time is below 2 seconds of *cpu*.

5 Conclusions

We have given a new explanation about the 3D scene perception. The principle of triangulation developped in this paper proves that the basic structure of a vision system is one camera with its motion and that the 3D scene perception can be efficiently achieved by a moving camera. This idea is applied to the problem concerning the reconstruction of a 3D polyhedral scene using a moving camera. We have dealed with the problems of : (a) the perception of 3D line segments and (b) the identification of 3D polygons.

We are now developping the presented work in the dynamic vision system VIDYR, especially the recursive estimation of 3D line segments within a sequence of images and the generation of the local motion favourable to the estimation of 2D dynamic primitives.

Acknowledgements

This research was supported by the project ORASIS, of the French National CNRS Program: PRC Communication Homme-Machine.

The authors would like to thank P.BOUTHEMY for the discussions about the extraction of the edgels' normal optical flows and M.L.BORGNE for helps about the use of QUATERNIONS theory to interpret the image corresponding velocities of camera.

Bibliographie

[ALO et al. 85] ALOIMONOS.J et SWAIN.M : " Shape From Texture ", *proceeding IJCAI85* , Los Angeles, Aug. 1985.

[ALO et al. 87] ALOIMONOS.J.Y et BANDYOPADHYAY.A : "Active Vision", *First International Conference On Computer Vision* ,June 8-11, 1987 London, England.

[AYA 88] AYACHE.N : "Construction et Fusion de Représentations Visuelles 3D - Application à la Robotique Mobile", *Thèse ès Science de l'Université de Paris-Sud, Centre d'Orsay*, 1988.

[BAL 81] BALLARD.D.H : " Generalizing The HOUGH Transform to Detect Arbitrary Shapes ", *Pattern Recognition Vol.13,No.2* , pp.111-122, 1981.

[BOL et al. 86] BOLLE.R.M et COOPER.D.B : "On optimaly combining pieces of information: with application to estimating 3D complex object position from range data", *IEEE, PAMI-8, No.5* , Septempber, 1986.

[BOR87] BORGNE.M.L : " QUATERNIONS ET CONTROLE SUR L'ESPACE DES ROTATIONS ", *Rapport Intern de l'IRISA*, No.377, Octobre, 1987.

[BOU 85] BOUTHEMY.P : " Estimation Of Edge Motion Based on Local Modelling ", *SPIE Vol.595, Computer Vision For Robotics* pp.162 - 169 , Cannes, December, 1985.

[BRO et al. 87] BROWN.C, ALOIMONOS.J, SWAIN.M, CHOU.P et BASU.A : " Texture, Contour, Shape, and Motion ", *Parttern Recognition Letters 5* , pp.151 - 169, February, 1987.

[CAS et al. 88] CASTELOW.D.A, MURRAY.D.W, SCOTT.G.L and BUXTON.B.F : "Matching Canny Edgels to Compute the Principal

Components of Optic Flow", *Image and Computer Vision Vo.6 No.2*, may, 1988.

[CER et al. 85] CERNUSCHI-FRIAS.B, BELHUM.P.N et COOPER.D.B : "Etimating and recognizing parameterized 3D objects using a moving cameras", *IEEE, International Conference on Robotics and Automation*, 1985.

[CHA et al. 88] CHAUMETT.F and RIVES.P : "Les différentes Calibrations Nécessaires à la Réalisation d'un Système de Vision Active", *Research Report to be published at IRISA/France.* (in french)

[DER 87] DERICHE.R : "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector", *International Journal of Computer Vision*, P167-187,1987.

[ESP et al. 87] ESPIAU.B et RIVES.P : " Closed-Loop Recursive Estimation Of 3D Features For a Mobile Vision System ", *IEEE, International Conference on Robotics and Automation* March 30 - Aril 3 , Raleigh, 1987.

[FAU 83] FAUGERAS.O : " Towards a Flexible Vision System ", *Robot Vision*, pp.129 - 143, 1983, Springer Verlag.

[HEG 87] HEGRON.G, ARNALDI.B and PRIOL.T : "VISYR : A simulation tool of synthetic vision for robotics", *Pro. of MARI87, Vo.II*, Paris, mai 1987.

[HOR 77] HORN.B.K.P : " Understanding Image Intensities ", *Artificial Intelligence.8(2)*, pp.201 - 231, 1977.

[IKE et al. 81] IKEUCHI.K et HORN.B.K.P : " Numerical Shape From Shading and Occluding Boundaries ", *Artificial Intelligence.17*, pp.409 - 460, 1981.

[IKE 84] IKEUCHI.K, "Shape from regular patterns", *Artifficial Intelligence 22*, 1984.

[LAU et al. 87] LAURENDEUA.D and POUSSART.D : "Model Building of Three-Dimensional Polyhedral Objects Using 3D Edge Information and Hemispheric Histogram", *IEEE, Journal of Robotics and Automation, Vo.3, No.5*, Oct, 1987.

[LOW 87] LOWE.D.G : "Three-Dimensional Object Recognition from Single Two-dimensional Images", *Artificial Intelligence 31* , 355 - 395, 1987.

[MCI et al. 88] MCINTOSH.J and MUTSH.K.M : "Matching Straight Lines", *Computer Vision, Graphics, and Images Processing 43* , Sept, 1988.

[MUR et al. 88] MURRY.D.W, CASTELOW.D.A and BUXTON.B.F : "From an image sequence to a recognized polyhedral objet", *Image and Computer Vision Vo.6 No.2*, May,1988.

[NEV et al. 80] NEVATIA.R and BABU.K.R : "Linear Feature Extraction and Description", *Computer Graphics and Image Processing 13* , 1980.

[OCH et al. 83] OCHIMA.M et SHIRAI.Y : " Objet Recognition Using Three Dimensional Information ", *IEEE, PAMI-5, No4* , July, 1983.

[RIC 85] RICHARDS.W : "Structure from stereo and motion", *J.opt.soc.Am.A, Vol.2, No.2* , February, 1985.

[RIV 87] RIVES.P : "Dynamic Vision: Theoretic Capability and Practical Problems" *Nato Workshop on Kinematic and Dynamic Issues in Sensor Based Control*, IL CIOCCO, TUSCANY Italy, October 25-31, 1987.

[SHA 86] SHARIAT.H : "The Motion Problem: How to Use More Than Two Frames ", University of Southern California, October 6, 1986.

[WAN et al. 87] WANG.Y.F, MITICHE.A et AGGARWAL.J.K : " Computation of Surface Orientation and Structure of Objets Using Grid Coding ", *IEEE TRANS. PAMI-9. No.1* , pp 129 -137, January, 1987.

[WAX et al. 86] WAXMAN.A.M and SINHA.S.S : "Dynamic Stereo: Passive Ranging to Moving Objects from Relative Image Flows", *IEEE TRANS. PAMI-8. No.4* , pp 129 -137, July, 1986.

[XIE et al. 88a] XIE.M and RIVES.P : "Identification of 3D Polygons from a set of 3D Line Segments", *Research Report No.427*, IRISA/France, Sept. 1988. (in french)

[XIE et al. 88b] XIE.M and RIVES.P : "From Edges to Line Segments: A Robust Algorithm", *Research Report No.429*, IRISA/France, Sept. 1988. (in french)

[XIE et al. 88c] XIE.M and RIVES.P : "An Accurate Algorithm for the Matching of 2D Line Segments within a Sequence of Images", *Research Report No.430*, IRISA/France, Oct. 1988. (in french)

[XIE et al. 88d] XIE.M and RIVES.P : "3D Vision: A New Method of Triangulation for the Monocular or Polynocular Vision within a Sequence of Images", *Research Report to be published at IRISA/France.* (in french)

[ZHU et al. 88] ZHUANG.X, HUANG.T.S, AHUJA.N and HARALICK.R.M : "A Simpified Linear Optic Flow-Motion Algorithm", *Computer Vision And Image Processing 42*, 1988.

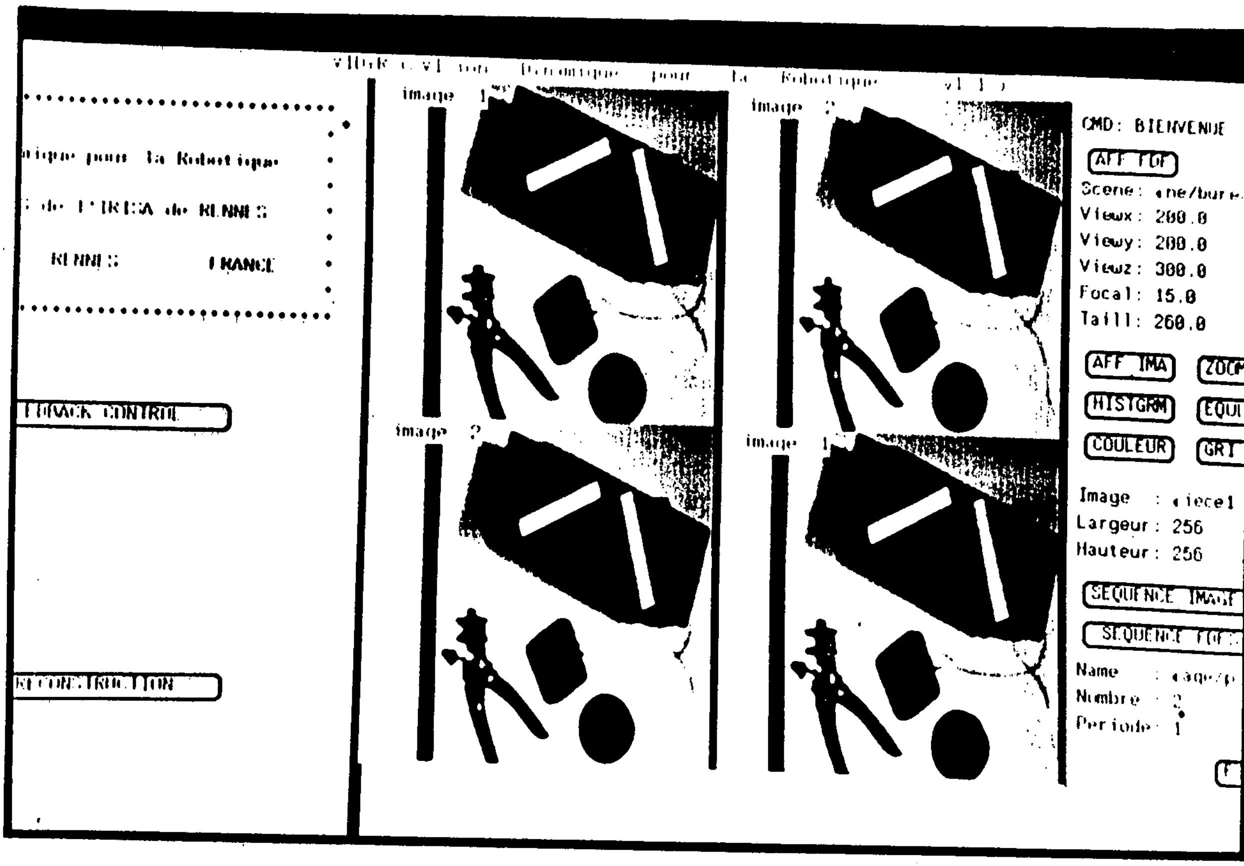


Figure 14 : The sequence of images used to show (a) the polygonal approximation of a edge image and (b) the matching of 2D line segments.

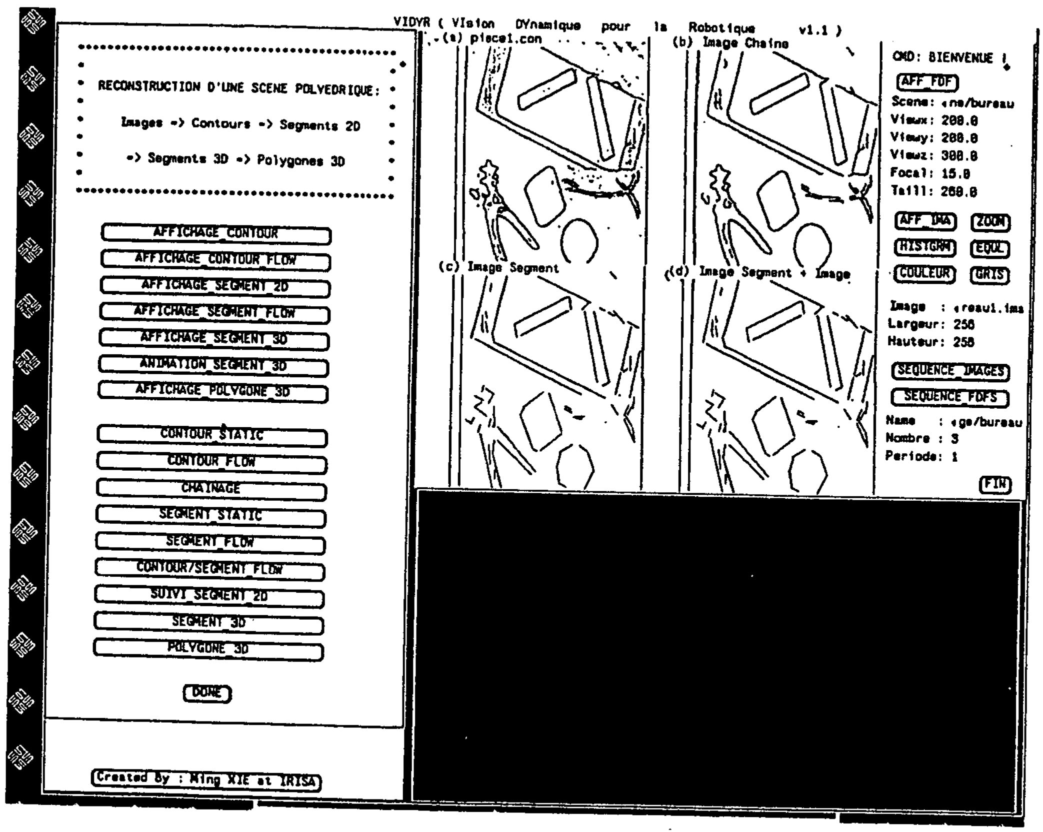


Figure 15 : An exemple of the polygonal approximation of a edge image.It is shown : (a) the edge image; (b) the image of chains; (c) and (d) the 2D line segments.

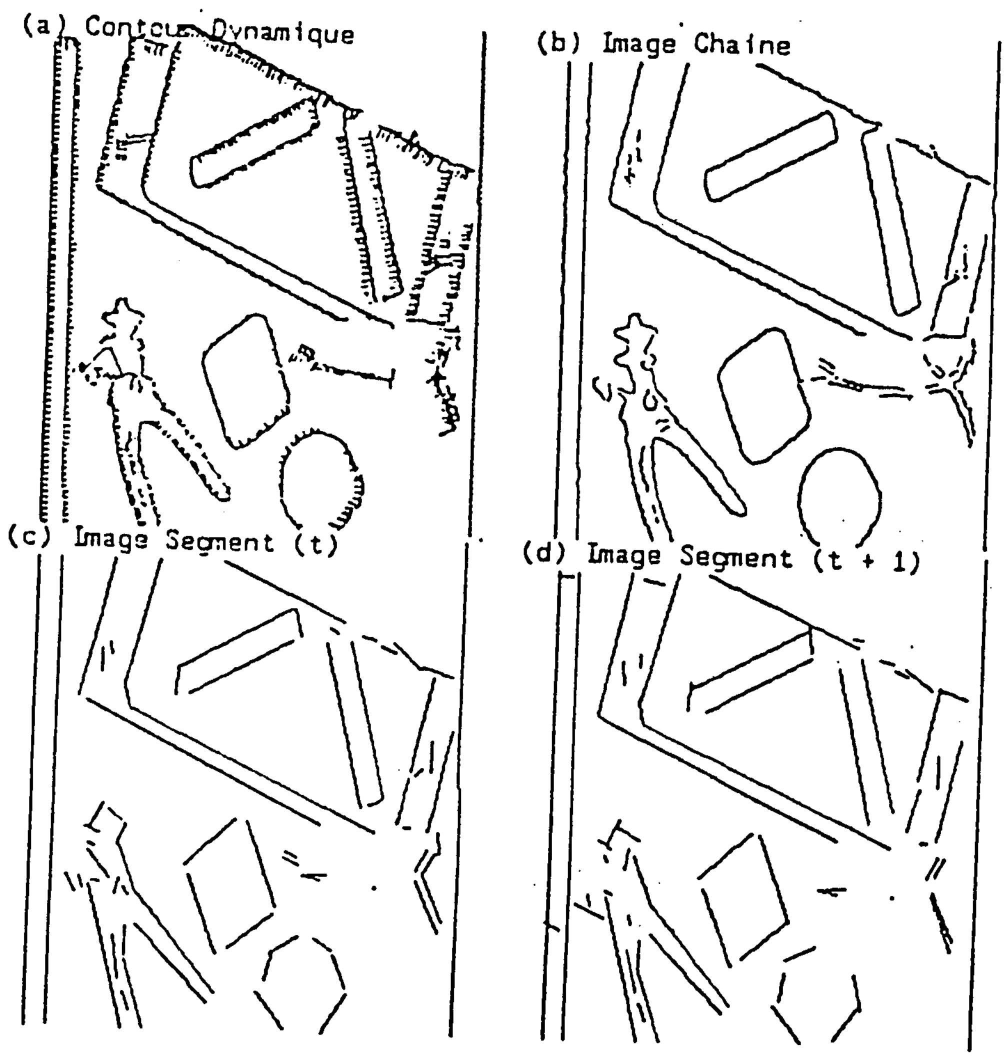


Figure 16 : The temporal matching of 2D line segments. It is shown : (a) the edgels' normal optical flow; (b) the edge image; (c) the estimated 2D line segments at the present image; (d) the corresponding predicted 2D line segments at the next image.

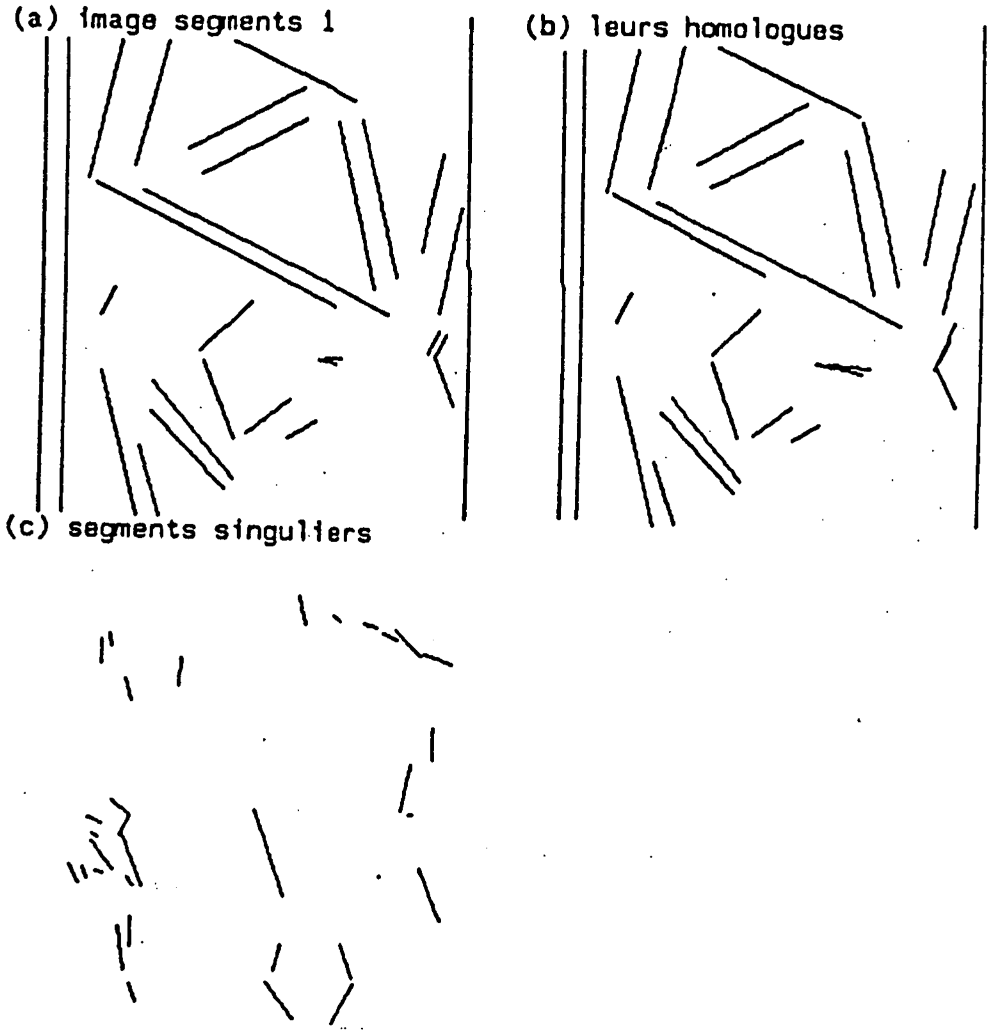


Figure 17 : The temporal/spatial matching of 2D line segments. It is shown : (a) the 2D line segments at the present image; (b) the corresponding 2D line segments at the next image and (c) the 2D line segments missed to be matched.

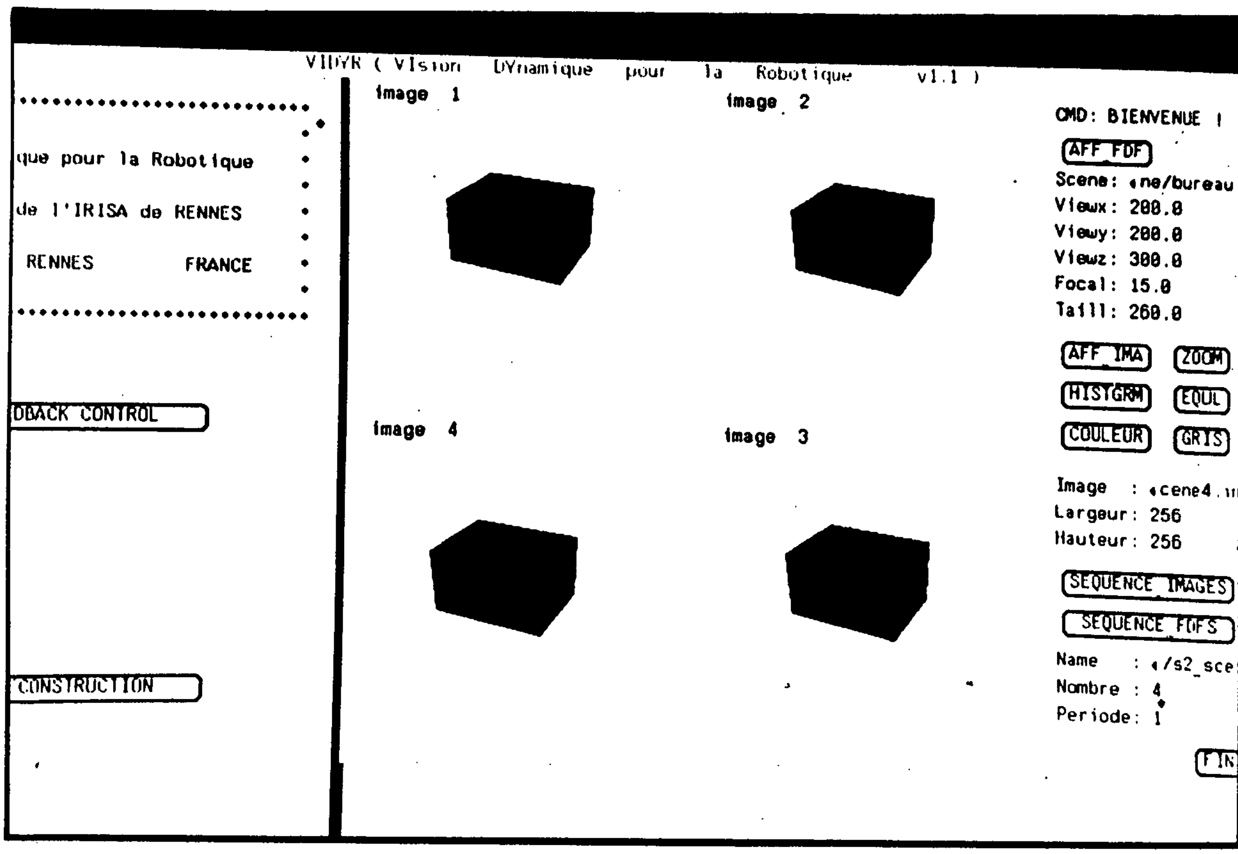


Figure 18 : The sequence of synthetic images used to show the triangulation of 3D line segments.

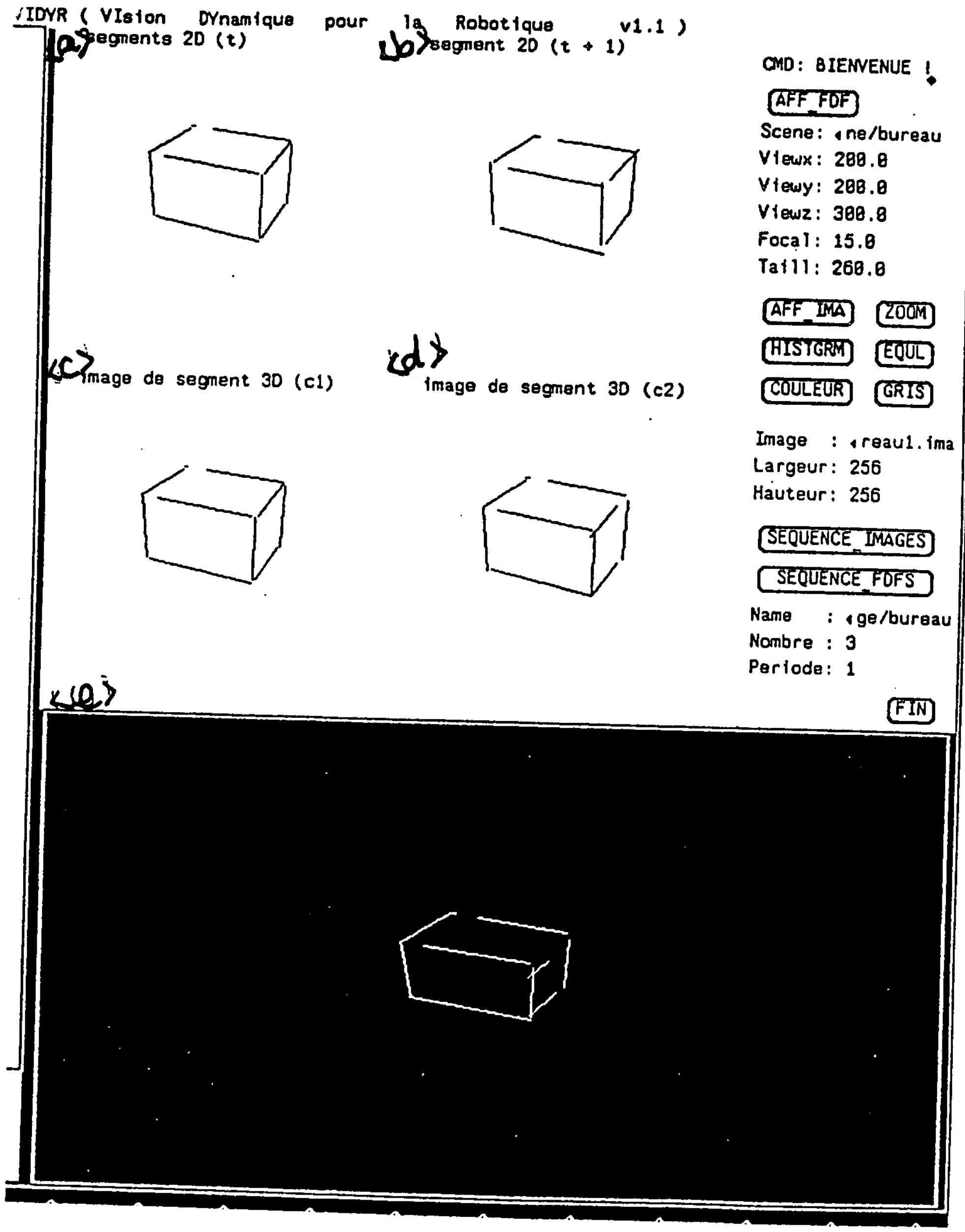


Figure 19 : The triangulation of 3D line segments: (a) and (b) the matched 2D line segments in two images; (c) and (d) the projection of the estimated 3D line segments to the camera at two different positions; The figure below is the 3D visualization of the 3D line segments.

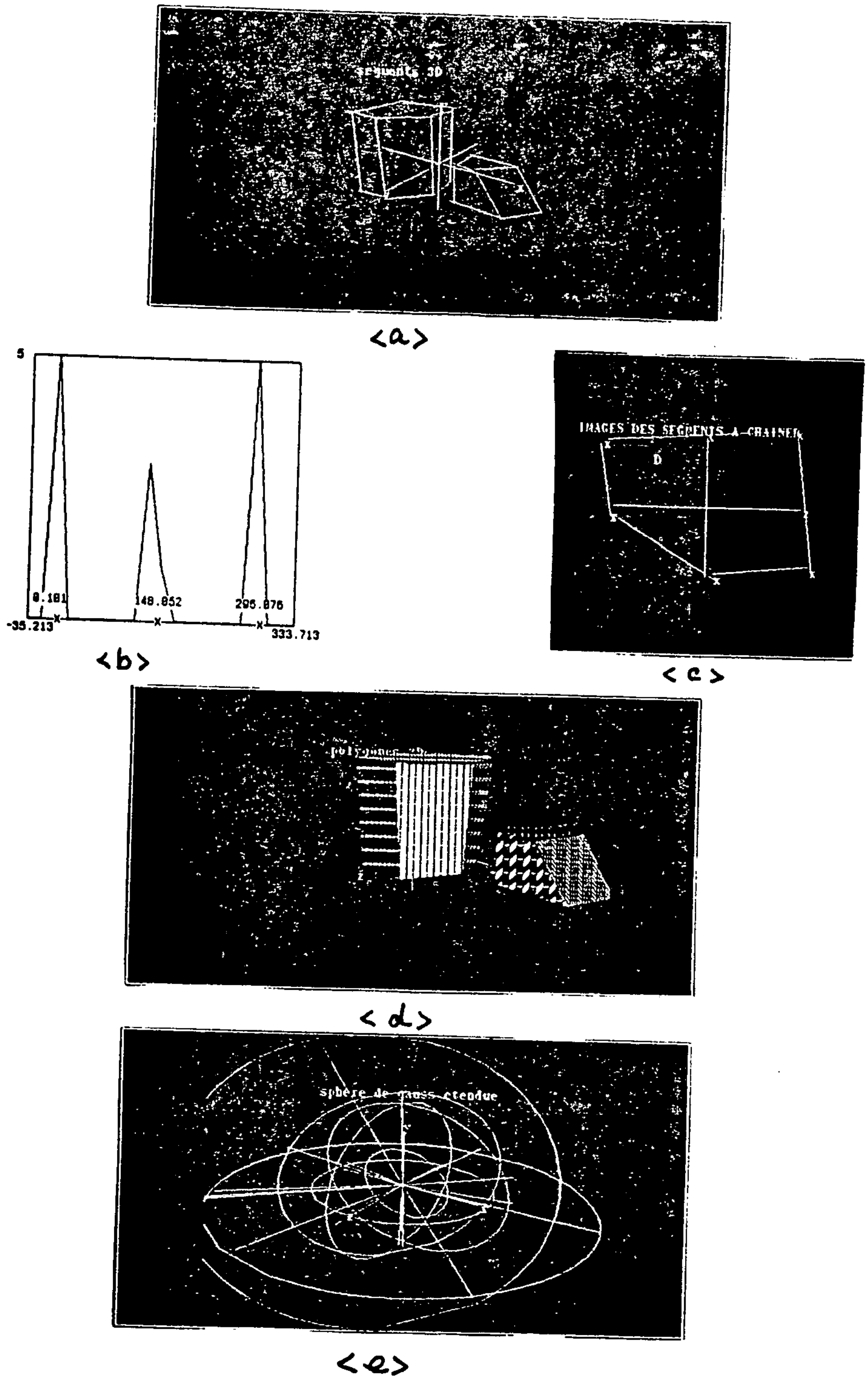


Figure 20 : The identification of 3D polygons. It is shown : (a) the set of 3D line segments with about 5% errors; (b) an exemple of the estimation of the parameters ρ;(c) an exemple of the linking of 3D line segments; (d) the estimated 3D polygons; (e) the representation of the estimated 3D polygons in the Extended GAUSS Sphere.

LISTE DES DERNIERES PUBLICATIONS INTERNES

PI 438 - A PROPOS DE LA RESOLUTION D'UN SYSTEME LINEAIRE DANS UN CORPS FINI : ALGORITHMES ET MACHINES PARALLELES
Hervé LE VERGE, Patrice QUINTON, Yves ROBERT, Gilles VILLARD
22 Pages, Novembre 1988.

PI 439 - ALPHA DU CENTAUR : A PROTOTYPE ENVIRONMENT FOR THE DESIGN OF PARALLEL REGULAR ALGORITHMS
Pierrick GACHET, Patrice QUINTON, Christophe MAURAS, Yannick SAOUTER
20 Pages, Novembre 1988.

PI 440 - CONSTRUCTION METHODIQUE D'UN ALGORITHME REPARTI DE DETECTION DE LA TERMINAISON
Jean-Michel HELARY, Michel RAYNAL
18 Pages, Décembre 1988.

PI 441 - LES GRAPHES A MOTIF
Didier CAUCAL
46 Pages, Décembre 1988.

PI 442 - CAUSAL TREES
Philippe DARONDEAU, Pierpaolo DEGANO
44 Pages, Décembre 1988.

PI 443 - TROIS IMPLANTATIONS DU RECUPERATEUR DE MEMOIRE DE LA MACHINE MALI
Michel LE HENAFF, Hervé SANSON
118 Pages, Décembre 1988.

PI 444 - ANALYSE FACTORIELLE LISSEE ET ANALYSE FACTORIELLE DES DIFFERENCES LOCALES
Brigitte ESCOFIER, Habib BENALI
34 Pages, Décembre 1988.

PI 445 - MULTISCALE STATISTICAL SIGNAL PROCESSING
Michèle BASSEVILLE, Albert BENVENISTE
16 Pages, Décembre 1988.

PI 446 - MODELES STATISTIQUES TEMPS-ECHELLE EN TRAITEMENT DU SIGNAL
Michèle BASSEVILLE, Albert BENVENISTE
60 Pages, Décembre 1988.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique